

新世纪高职高专规划教材·计算机专业基础系列

计算机专业基础系列

悦语言程序设计 基础与应用

李铮摇叶艳冰摇王德俊摇编著



清华大学出版社

北京

前 言

C 语言是目前国内外广泛流行的一种计算机程序设计语言。它具有高效、简洁和接近汇编语言的特点,既适合编写系统程序,又适合编写应用程序。C 语言自问世以来,以其独特的魅力吸引了广大计算机用户学习和使用。

本书在编写的内容和形式上体现了高职高专教育的培养目标——“培养适合生产、建设、管理、服务第一线需要的高等技术应用性专门人才”,以及高职高专教育要求知识面宽、基本理论和原理知识适度并加强实际技能培养等。在编写教材时,突出讲解 C 语言中最基本、最常用的内容,重点放在语言本身的一些难点和程序设计方法上,力求简洁、明确、清晰,使读者能够较快入门,并逐步学会自己动手编写程序。

全书分为两大部分,共 13 章。第一部分为基础篇,即前 10 章。第 1 章到第 9 章对 C 语言的内容进行了讲解,第 10 章通过一个示例对 C 语言的内容进行了综合应用,学会如何完成一个系统的分析和设计。基础篇的每一章开始处都安排了本章的基本内容提要 and 基本要求,使读者有明确的阅读方向;在结束处总结了初学者常见的问题和错误,并对这些问题和错误进行了分析和纠正,在此基础上对全章进行总结。第二部分为应用篇,即第 11 章到第 13 章,主要介绍了 C 语言的几种开发工具,以及在图形用户接口(GUI)和网络编程方面的应用。

本书还为读者提供了丰富的附录内容,特别是在附录 D 中通过实例较详细地讲解了程序的调试方法,相信会为读者提供有益的帮助。

本书由李铮担任主编,叶艳冰、王德俊担任副主编。李铮编写第 1~4,7~9,11~13 章,叶艳冰编写第 5、6 章,王德俊编写第 10 章。全书由李铮统阅定稿。上海交通大学技术学院刘远生教授在本书的编写过程中提出了许多宝贵的建议和修改意见,在此表示衷心的感谢。

由于作者水平有限,书中难免存在缺点和不足之处,恳请各位专家、老师和同学提出宝贵意见(作者 E-mail: LE.cpp@163.com)。本书的所有程序代码均已调试通过,读者可以上网或 E-mail 免费索取。网址: <http://www.tup.tsinghua.edu.cn>。E-mail: gzgz@tup.tsinghua.edu.cn.

编 者

2005 年 1 月于上海交通大学

出版说明

高职高专教育是我国高等教育的重要组成部分,担负着为国家培养并输送生产、建设、管理、服务第一线高素质技术应用型人才的重任。

进入 21 世纪后,高职高专教育的改革和发展呈现出前所未有的发展势头,其学生规模已占我国高等教育的半壁江山,成为我国高等教育的一支重要的生力军;办学理念上,“以就业为导向”成为高等职业教育改革与发展的主旋律。近两年,教育部召开了三次产学研交流会,并启动四个专业的“国家技能型紧缺人才培养项目”,同时成立了 35 所示范性软件职业技术学院,进行两年制教学改革试点。这些举措都表明国家正在推动高职高专教育进行深层次的重大改革,向培养生产、服务第一线真正需要的应用型人才的方向发展。

为了顺应当前我国高职高专教育的发展形势,配合高职高专院校的教学改革和教材建设,进一步提高我国高职高专教育教材质量,在教育部的指导下,清华大学出版社组织出版“21 世纪高职高专规划教材”。

为推动规划教材的建设,清华大学出版社组织并成立“高职高专教育教材编审委员会”,旨在对清华版的全国性高职高专教材及教材选题进行评审,并向清华大学出版社推荐各院校办学特色鲜明、内容质量优秀的教材选题。教材选题由个人或各院校推荐,经编审委员会认真评审,最后由清华大学出版社出版。编审委员会的成员皆为教改成效大、办学特色鲜明、师资实力强的高职高专院校、普通高校以及著名企业,教材的编写者和审定者都是从事高职高专教育第一线的骨干教师和专家。

编审委员会根据教育部最新文件政策,规划教材体系,比如部分专业的两年制教材;“以就业为导向”,以“专业技能体系”为主,突出人才培养的实践性、应用性的原则,重新组织系列课程的教材结构,整合课程体系;按照教育部制定的《高职高专教育基础课程教学基本要求》,教材的基础理论以“必要、够用”为度,突出基础理论的应用和实践技能的培养。

本套规划教材的编写原则如下:

- (1) 根据岗位群设置教材系列,并成立系列教材编审委员会;
- (2) 编审委员会规划教材、评审教材;
- (3) 重点课程进行立体化建设,突出案例式教学体系,加强实训教材的出版,完善教学服务体系;

- (4) 教材编写者由具有丰富教学经验和多年实践经验的教师共同组成,建立“双师型”编者体系。

本套规划教材涵盖了公共基础课、计算机、电子信息、机械、经济管理以及服务等大类的主要课程,包括专业基础课和专业主干课。目前已经规划的教材系列名称如下:

公共基础课

公共基础课系列

计算机类

计算机基础教育系列

计算机专业基础系列

计算机应用系列

网络专业系列

软件专业系列

电子商务专业系列

电子信息类

电子信息基础系列

微电子技术系列

通信技术系列

电气、自动化、应用电子技术系列

机械类

机械基础系列

机械设计与制造专业系列

数控技术系列

模具设计与制造系列

经济管理类

经济管理基础系列

市场营销系列

财务会计系列

企业管理系列

物流管理系列

财政金融系列

服务类

旅游系列

艺术设计系列

本套规划教材的系列名称根据学科基础和岗位群方向设置,为各高职高专院校提供“自助餐”形式的教材。各院校在选择课程需要的教材时,专业课程可以根据岗位群选择系列;专业基础课程可以根据学科方向选择各类的基础课系列。例如,数控技术方向的专业课程可以在“数控技术系列”选择;数控技术专业需要的基础课程,属于计算机类课程可以在“计算机基础教育系列”和“计算机应用系列”选择,属于机械类课程可以在“机械基础系列”选择,属于电子信息类课程可以在“电子信息基础系列”选择。依此类推。

为方便教师授课和学生学习,清华大学出版社正在建设本套教材的教学服务体系。本套教材先期选择重点课程和专业主干课程,进行立体化教材建设:加强多媒体教学课件或电子教案、素材库、学习盘、学习指导书等形式的制作和出版,开发网络课程。学校在选用教材时,可通过邮件或电话与我们联系获取相关服务,并通过与各院校的密切交流,使其日臻完善。

高职高专教育正处于新一轮改革时期,从专业设置、课程体系建设到教材编写,依然是新课题。希望各高职高专院校在教学实践中积极提出意见和建议,并向我们推荐优秀选题。反馈意见请发送到 E-mail:gzgz@tup.tsinghua.edu.cn。清华大学出版社将对已出版的教材不断地修订、完善,提高教材质量,完善教材服务体系,为我国的高职高专教育出版优秀的高质量的教材。

高职高专教育教材编审委员会

目 录

第 1 篇 基础篇

第 1 章 悦语言概论	猿
1.1 悦语言的简单历史	猿
1.2 结构化程序设计	源
1.2.1 算法和程序	源
1.2.2 结构化程序设计的思想和方法	源
1.3 编写一个简单的 悦语言程序	远
1.3.1 程序设计的一般方法	远
1.3.2 悦语言程序的一般组成	愿
1.3.3 程序的质量	怨
1.3.4 书写程序时应遵循的规则	怨
1.3.5 在 悦语言环境下执行一个 悦语言程序的基本步骤	员
1.3.6 学好 悦语言的一些建议	员
1.4 小结	员
习题	员
实验与实训	猿
第 2 章 悦语言基本数据类型和运算	源
2.1 悦语言的基本数据类型	源
2.1.1 悦语言的描述符	源
2.1.2 悦语言的基本数据类型	苑
2.2 运算符和表达式	苑
2.2.1 基本运算符和表达式	苑
2.2.2 数据类型间的转换规则	猿
2.3 标准输入输出语句	猿

猿源猿标准输出语句及其说明	猿猿
猿源猿标准输入语句及其说明	猿猿
猿源猿常见错误分析	猿源
猿源猿数据类型和变量说明中的错误分析	猿源
猿源猿标准输入输出语句使用中的错误分析	猿猿
猿源猿小结	猿源
习题	猿猿
实验与实训	猿源
第 猿章 猿程序流程的控制	猿怨
猿猿猿概述	猿怨
猿猿猿控制语句	猿怨
猿猿猿猿条件语句	猿怨
猿猿猿猿例题与分析	猿怨
猿猿猿开关语句	猿怨
猿猿猿循环语句	猿怨
猿猿猿猿增量语句	猿怨
猿猿猿猿减量语句	猿怨
猿猿猿猿本语句	猿怨
猿猿猿猿逻辑表达式和 早断语句	猿怨
猿猿猿猿程序举例	猿怨
猿猿猿常见错误分析	猿怨
猿猿猿小结	猿怨
习题	猿怨
实验与实训	猿怨
第 源章 猿函数	猿怨
源源猿函数的定义	猿怨
源源猿猿函数使用的意义和分类	猿怨
源源猿猿函数定义形式	猿怨
源源猿函数的调用	猿怨
源源猿猿函数调用的形式	猿怨
源源猿猿函数调用的方式	猿怨
源源猿猿函数的嵌套调用	猿怨

源函数函数的递归调用	怨
源函数函数的参数	怨
源函数函数的实参和形参	怨
源函数函数的返回值	怨
源函数例题与分析	怨
源常见错误分析	怨
源函数函数定义中的错误与分析	怨
源函数函数设计与使用中的错误与分析	怨
源小结	源
习题	源
实验与实训	源
第 缘章 数组和指针	源
缘一维数组	源
缘一维数组的定义	源
缘一维数组的初始化	源
缘一维数组的使用	源
缘一维数组作为函数参数	源
缘二维数组	源
缘二维数组的定义	源
缘二维数组的初始化	源
缘二维数组的使用	源
缘二维数组作为函数参数	源
缘字符数组	源
缘字符数组的定义和初始化	源
缘字符数组的输入和输出	源
缘字符串处理函数	源
缘字符串处理函数的应用	源
缘指针	源
缘指针及指针变量的定义	源
缘指针的简单用法	源
缘指针和数组	源
缘指针与数组的关系	源
缘指针数组	源

第 4 章 指针和函数	4.1
4.1.1 指向变量的指针作为函数参数	4.1.1
4.1.2 指向数组的指针作为函数参数	4.1.2
4.1.3 例题与分析	4.1.3
4.1.4 动态分配空间	4.1.4
4.1.5 常见错误分析	4.1.5
4.1.5.1 数组定义中的错误分析	4.1.5.1
4.1.5.2 数组赋值中的错误分析	4.1.5.2
4.1.5.3 数组使用中的错误分析	4.1.5.3
4.1.6 小结	4.1.6
习题	4.1.6
实验与实训	4.1.6
第 5 章 结构体和联合	5.1
5.1.1 结构	5.1.1
5.1.1.1 结构的意义和定义	5.1.1.1
5.1.1.2 结构体的简单使用	5.1.1.2
5.1.2 联合	5.1.2
5.1.2.1 联合的意义和定义	5.1.2.1
5.1.2.2 联合的简单使用	5.1.2.2
5.1.3 常见错误分析	5.1.3
5.1.3.1 结构和联合定义中的错误与分析	5.1.3.1
5.1.3.2 结构使用中的错误与分析	5.1.3.2
5.1.3.3 联合使用中的错误与分析	5.1.3.3
5.1.4 小结	5.1.4
习题	5.1.4
实验与实训	5.1.4
第 6 章 各种存储类的区别	6.1
6.1.1 概述	6.1.1
6.1.2 自动变量	6.1.2
6.1.2.1 定义方式	6.1.2.1
6.1.2.2 作用域和生存期	6.1.2.2
6.1.2.3 初始化	6.1.2.3

第 1 章 摇 寄存器变量	1
第 2 章 摇 外部变量	2
第 2.1 节 摇 定义方式	2
第 2.2 节 摇 作用域和生存期	2
第 2.3 节 摇 初始化	2
第 3 章 摇 静态变量	3
第 3.1 节 摇 内部静态变量	3
第 3.2 节 摇 外部静态变量	3
第 4 章 摇 内部函数和外部函数	4
第 5 章 摇 运行一个多文件的程序	5
第 6 章 摇 小结	6
习题	6
实验与实训	6
第 7 章 摇 预处理	7
第 7.1 节 摇 概述	7
第 7.2 节 摇 宏定义	7
第 7.2.1 节 摇 无参数宏	7
第 7.2.2 节 摇 带参数宏	7
第 7.3 节 摇 文件包含	7
第 7.4 节 摇 条件编译	7
第 7.5 节 摇 运行一个多文件的程序	7
第 7.6 节 摇 常见错误分析	7
第 7.7 节 摇 小结	7
习题	7
实验与实训	7
第 8 章 摇 文件	8
第 8.1 节 摇 概述	8
第 8.1.1 节 摇 文件的概念和分类	8
第 8.1.2 节 摇 缓冲文件系统和非缓冲文件系统	8
第 8.2 节 摇 缓冲文件系统的文件操作	8
第 8.2.1 节 摇 文件打开和关闭	8
第 8.2.2 节 摇 文件的读写	8

4.1.1 文件的定位	411
4.1.2 文件的检测	412
4.1.3 非缓冲文件系统的文件操作	413
4.1.4 常见错误分析	414
4.1.5 小结	415
习题	416
实验与实训	417
第 4 章 综合应用实例——学生简易选课管理系统	418
4.1 系统功能分析	418
4.1.1 总体功能分析	418
4.1.2 模块功能细分	419
4.2 主要数据类型定义	420
4.3 主要函数说明	421
4.4 主要源代码	422
4.5 运行实例说明	423
4.6 小结	424
实验与实训	425

第 5 篇 应用篇

第 5 章 图形处理	426
5.1 概述	426
5.2 图形函数	427
第 6 章 悦语言集成开发环境介绍	428
6.1 裁悦简介	428
6.1.1 裁悦的安装	429
6.1.2 裁悦的运行	430
6.1.3 裁悦主菜单	431
6.1.4 裁悦的配置	432
6.2 裁悦集成开发环境简介	433
6.2.1 灾悦简介	434
6.2.2 灾悦集成开发环境	435

参考文献.....

第 1 篇

基 础 篇

第 1 章 摇悦语言概论

第 2 章 摇悦语言基本数据类型和运算

第 3 章 摇悦程序流程的控制

第 4 章 摇悦函数

第 5 章 摇悦数组和指针

第 6 章 摇悦结构体和联合

第 7 章 摇悦各种存储类的区别

第 8 章 摇悦预处理

第 9 章 摇悦文件

第 10 章 摇悦综合应用实例——学生简易选课管理系统

C 语言概论

本章内容提要及要求

本章简要介绍了 悦语言的历史和特点 ,描述了程序、算法和结构化程序设计等概念。重点要求了解 悦语言函数的构成 ,以及完成一个程序设计的主要步骤和实现方法 ,并且了解代码书写规范 ,在书写程序时逐步养成良好的设计风格。

1.1 悦语言的简单历史

悦语言是目前世界上流行、使用最广泛的高级程序设计语言。早期的 悦语言主要用于 操作系统。由于 悦语言的强大功能和各方面的优点逐渐为人们所认识 ,20 世纪 60 年代 ,悦语言开始进入其他操作系统 ,并很快在各类大、中、小和微型计算机上得到了广泛的使用 ,成为当代最优秀的程序设计语言之一。同时 ,悦语言还具有绘图能力强、可移植性好、数据处理能力强等特点 ,故亦可用于三维、二维图形和动画的设计。

1.1.1 悦语言的简单历史

悦语言在 1969 年由美国电话电报公司 (AT&T) 贝尔实验室正式发表。同时 ,由 1969 年 1 月 1 日出版和 1970 年 1 月 1 日出版合著的《C 语言》一书对 悦语言做了详细的描述。在此之后 ,由美国国家标准学会 (ANSI) 制定了一个 悦语言标准 ,于 1983 年发表 ,通常称之为 标准悦语言。

在 悦的基础上 ,1983 年又由贝尔实验室的 1983 年 1 月 1 日出版推出了 悦语言。悦语言进一步扩充和完善了 悦语言 ,成为一种面向对象的程序设计语言。

1.1.2 悦语言的主要特点

悦语言是一种结构化语言。它层次清晰 ,便于按模块化方式组织程序 ,易于调试和维护。它的主要特点如下 :

- 既有高级语言的程序思想和设计方法 ,又有低级语言的操作能力。所以它也被称为“ 中级语言 ”。

- 结构化的体系结构。其层次清晰,便于按模块化方式组织程序,易于调试和维护。
- 强大的处理能力。它不仅具有丰富的运算符和数据类型,便于实现各类复杂的数据结构,还可以直接访问内存的物理地址。
- 广泛的可移植性,可以方便地移植到不同的软、硬件环境。
- 代码效率高。
- 运算符优先级太多,类型转换限制少,检验较弱,不够安全。

结构化程序设计

算法和程序

程序

通俗地说,程序就是完成某项任务或事务的一种既定方式和过程。

事实上,在日常生活中的许多事情都有一定的程序。例如,我们开车需要经历下面几个有顺序的动作方可完成:

- 准备好车钥匙;
- 找到停车的位置;
- 进入车厢;
- 用钥匙打开车锁;
- 发动车子;
- 开车出行。

当然,对每个动作还可以根据需要进行进一步细分。

算法

所谓算法,就是为完成某项任务或事务而采用的方法和步骤。

就上面开车的例子而言,为达到开车出行的目的,需要按顺序经过若干步骤才能实现,并且在每个步骤中还要将它的实现方法考虑进去。例如,如何准备好车的钥匙,如何找到停车的位置等。

算法可以有多种描述方式,例如,上面采用的是自然语言的描述方法等。

程序设计

程序设计就是人们编制计算机程序的工作。

结构化程序设计的思想和方法

结构化程序设计思想

结构化程序设计的基本含义是指一个系统由层次化的程序模块构成,每一个模块只

有一个入口和出口,每一个模块只归某个上级模块调用,有模块联结的准则和构造模块的标准,并且用系统结构图来表达系统的结构,尽可能用最优的方式将系统内各个部分组织起来,而不是用若干个程序去拼凑。

1.1 结构化程序设计方法

(1) 自顶向下

程序设计时,应先考虑总体后考虑细节,先考虑全局目标后考虑局部目标。即先对问题进行仔细分析,确定其输入、输出数据,写出程序运行的主要过程和任务;然后从大的功能方面把一个问题的解决过程分成几个子问题,每个子问题形成一个模块。

(2) 逐步细化

对于复杂问题,先设计分解成一些子问题,然后逐步细节化,直到整个问题可以用程序设计语言明确地描述出来为止。

(3) 模块化设计

把一个较大的程序划分为若干子程序,每一个子程序模块解决一个独立的功能,每一个子模块又可根据需要继续划分为更小的子模块使程序具有层次结构。

需要注意的是运用这种编程方法,考虑问题必须先进行整体分析,避免边写边想。

(4) 结构化编码

结构化编码的显著特点是代码和数据的分隔化,即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰,便于使用、维护和调试。

下面以求解 10 个整型数据的平均值为例,说明结构化程序设计的思想和方法。

首先根据题目要求确定最终的目标,即求解 10 个整型数据的平均值。为了达到这个目标,必须先确定一系列子目标:如何得到这 10 个数?如何计算出平均值?再如何显示它的平均值?事实上就是把这个最终目标进行了分解。当解决了所有子目标后,最终目标也就实现了。而每一个子目标又需要再进行仔细考虑,即根据需要逐步细化。分析和分解的结果可用方框图的方式来表达,见图 1.1。

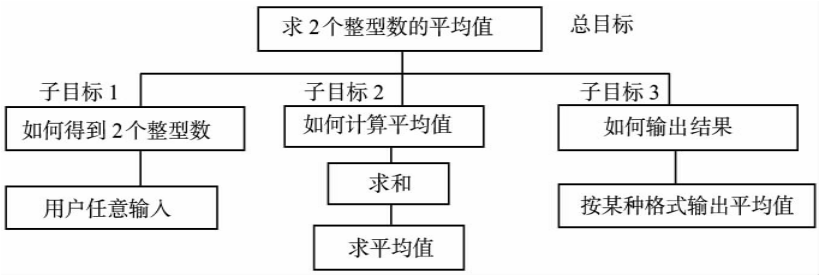


图 1.1 求解 10 个整型数据平均值的系统框图

摇摇编写一个简单的悦语言程序

摇摇程序设计的一般方法

摇摇在这一节中,通过实例向读者阐明一个简单的程序是如何设计出来的,使读者对程序设计有一个初步的认识,而不仅仅是程序结构和代码本身。当然,要想真正地掌握还要通过后面的深入学习,在这里首先是要了解设计的过程和思想。

前面已经阐述了结构化程序设计的方法,一般来说完成一个程序的设计主要包括下面源个步骤。

- 分析问题。
- 构造问题模型及求解步骤。
- 模块的代码实现。
- 调试与优化代码,完善程序。

这里仍以计算圆个整型数的平均值为例,说明这源个设计步骤。

源.1 分析问题

在这一步骤中,首先要分析出问题最后需要输出的是什么(即需求是什么),已知的输入又是什么(即所给的条件是什么)。也就是说,首先要把问题弄得更清楚、更明确些,然后据此定义出表示输入和输出的变量。就上面提出的例子而言:

输入摇圆个待计算的整型数据,使用变量 x_1 和 x_2 表示(这是由自己定义的)。

输出摇圆个整型数的平均值,使用变量 avg 表示(这也是由自己定义的)。

其次要寻找一种方法实现从输入到输出的转换,也就是找到一种方法,经过有限步的处理能从输入中获取需要的结果并输出。这也就是前面提到的,如何将总目标分解成一系列子目标,再一一解决这些子目标,最后得解。需要说明的是,人们解决问题的方法是多种多样的,程序设计并没有一个标准的答案,只是追求一种相对比较合理、比较圆满的解决问题的途径而已。

就上例而言,使用的方法是首先计算圆个整型数据的和,再得其平均值。

源.2 构造问题基本模型和求解步骤

这一步骤实际上是把第一步骤中的分析工作更加细化,把复杂的问题分解为相对简单的问题然后再逐步细化。也就是将解决问题的过程按顺序一步一步地分解出来。分解和描述如图 员.1 所示。

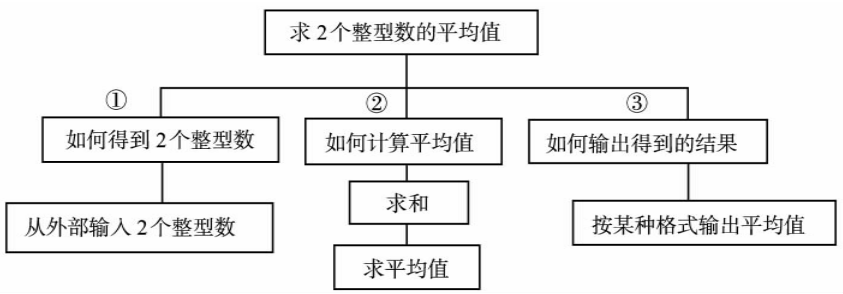


图 1.1 求解两个整型数据平均值的系统分解框图

模块的代码实现

根据前一步中分解出来的模块分别进行设计。

(1) 如何得到 2 个整型数据：从外部输入 2 个整型数据

为了使程序具有通用性，设定程序可以通过键盘输入任意 2 个整型数。根据该功能要求，考虑可以直接使用悦语言提供的标准输入函数来实现。

```
int a, b;
scanf("%d %d", &a, &b);
```

(2) 如何计算平均值：先求和，再求平均值

① 先求和。我们用一个变量 sum 来保存 2 个整型数相加的和：

```
int sum;
sum = a + b;
```

② 再求平均值(以实数形式表示该平均值)：

```
float avg;
avg = sum / 2;
```

(3) 如何输出得到的结果：可以按某种格式输出平均值

根据这个功能需求，同样可以直接使用悦语言提供的标准输出函数以实数形式输出该平均值：

```
printf("Average: %f", avg);
```

在完成了每个细节模块的编写之后，将各个部分连接成一个完整的程序雏形。事实上，一般的程序从宏观上看它们都是由三段式构成的，如图 1.2 所示。

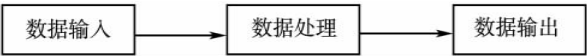


图 1.2 程序三段式